Participatory Design in Information Systems Development

Bødker, Keld; Kensing, Finn; Simonsen, Jesper

Published in:
Reframing Humans in Information Systems Development

Publication date:
2011

Document Version
Peer reviewed version

Citation for published version (APA):
Bødker, K., Kensing, F., & Simonsen, J. (2011). Participatory Design in Information Systems Development. In H. Isomäki, & S. Pekkola (Eds.), *Reframing Humans in Information Systems Development* (pp. 115-134). Springer. http://www.springer.com/computer/hci/book/978-1-84996-346-6

Chapter 7
Participatory Design in Information Systems Development

Keld Bødker, Finn Kensing, and Jesper Simonsen

7.1 Introduction

This chapter deals with IT design in an organizational setting – be it a medium sized service company, a large industrial company, a small entrepreneurial knowledge company, or a public company or institution. In such work settings we often find a complex organizational structure, including several management levels, diverse professional groups, workplace cultures, and established working relations where new IT projects challenge the established ways-of-working. This is also the domain of 'classic' information systems development (ISD) approaches. This chapter presents the principles, key ideas, and experiences from using the participatory design method known as the 'MUST method'[1], developed by the authors (Bødker et al. 2004).

Iivari et al. (2009) identify three fields where system development has been the topic of research: Software Engineering (SE), Information Systems (IS), and Human Computer Interaction (HCI). User involvement has not been the topic of research in SE, but the topic is well researched in IS and HCI. However, Iivari et al. (2009) note as an important aspect that the literature is not clear on *how* user involvement should be integrated with current approaches to system development: While agile methods include ways of incorporating customer requirements, Ballejos and Montagna (2008) demonstrate that they fail to support stakeholder identification; DeMichelis et al. (1998) show limitations towards the design and management of organizational change; and Coughlan and Macredie (2002) illustrate insufficiencies in negotiations of requirements between different stakeholder and user groups.

[1] MUST is a Danish acronym for initial participatory design activities.

K. Bødker (✉) and J. Simonsen
Roskilde University, Denmark
e-mail: keldb@ruc.dk; simonsen@ruc.dk

F. Kensing
University of Copenhagen, Denmark
e-mail: kensing@citi.ku.dk

H. Isomäki and S. Pekkola (eds.), *Reframing Humans in Information Systems Development*, Computer Supported Cooperative Work 201,
DOI 10.1007/978-1-84996-347-3_7,

In this chapter, we describe the participatory design approach, presented by Bødker et al. 2004, illustrating *how* to engage *which* users in *what* type of design activities.

The MUST method is a 'meta-method' providing resources that has to be applied specifically to a situated IT design project. The method provides four types of resources: *Concepts* for the designer to understand and frame the situation, *principles* forming the backbone of the method, suggestions for how to *organize* the design project, and finally, a *toolbox* of techniques and presentation tools to support the various activities, see Fig. 7.1.

In the remaining part of the introductory section our interpretation of the concept of *design* is discussed. Section 7.2 discusses the most central concept – the user. *Who* is the user, and how can we identify and include them in design activities? Section 7.3 describes the four *principles* that form the backbone of the MUST method – illustrated by small vignettes from industrial projects. We use the vignettes to illustrate condensed experience describing typical situations or challenges faced in specific situations. Each vignette is identified by a number. Space does not allow for a thorough description of how to organize the design project according to clearly identified decision points, let alone the techniques and tools. For further details please see (Bødker et al. 2004). Section 7.4 compares the MUST method to other methods and approaches. Section 7.5 concludes the chapter by discussing implications for industry, education, and research.

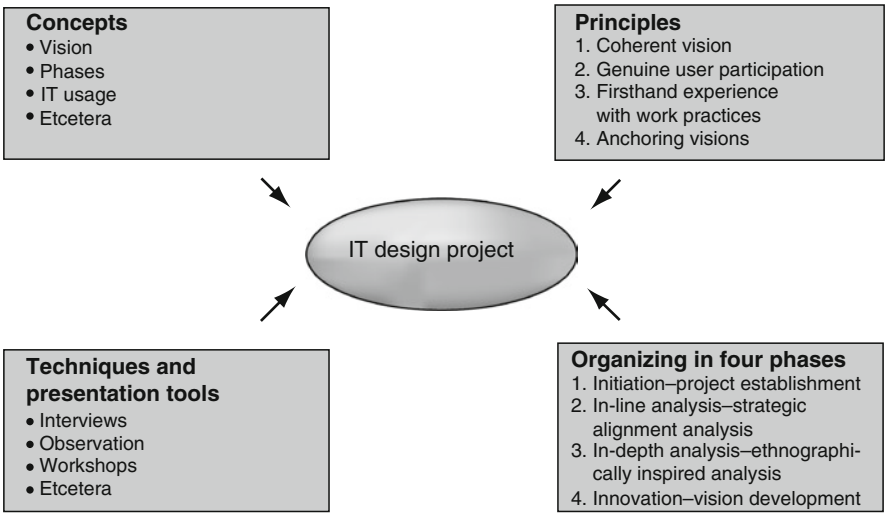

Fig. 7.1 Four types of resources for IT design using the MUST method[2]

[2] Bødker, Keld, Finn Kensing, and Jesper Simonsen, Participatory IT Design: Designing for Business and Workplace Realities, Figure 1.3 © 2004 Massachusetts Institute of Technology, by permission of The MIT Press.

7.2 What Is IT Design?

Our understanding of IT design and design activities in relation to an overall IT project is inspired by better established design traditions, such as architecture. In architecture architects analyze clients' wishes and needs, designing a building's form and function over several iterations taking an appreciation of the context into account. At first they design at a general and conceptual level, then in greater detail – in order to develop design ideas and prepare the construction process.

In the construction industry, traditions and experience have developed over centuries, stipulating how the construction process is conducted and how architects cooperate with the client and the many trade groups that, at one time or another, are drawn into a construction project. Over many years, a wealth of experiences has been gained and overall standards have been established in many areas of the construction process for example regarding division of work, as well as in terms of the content and detail of various specifications developed in the process. Traditions are less developed in the IT world. A marked difference is the construction industry's established interfaces between phases, enabling calls for tenders to be issued at several different points and with varying scope. For example, many large projects start by pre-qualifying a small number of architects who then for a small budget is asked to develop their design for an architectural competition. The winner is then awarded the project, and more detailed design is carried out. At a later stage another call for tenders is opened based on a detailed design for the construction, and the winner is awarded the contract for the construction of the building.

We understand IT design as similar to the activities of architects in the construction industry: Initial activities of analysis and design that outline visions for future IT usage and support decisions about which visions best meet business goals and user needs for IT support in their work. An IT design project thus constitutes an important element in an organizational clarification process leading to visions for one or more sustainable uses of IT. An explicit upfront design component of an IT project is important. This can be realized in many ways. In most projects, it is necessary to divide an IT project into a design project and an implementation project, separated by a call for tenders, see Fig. 7.2. In fact, this is the case with all large public IT projects in Europe according to the European Union rules for invitations to tender and award contracts. This model is used as a reference for identifying the IT design project throughout this chapter. The MUST method, including its techniques and presentation tools, may be used in other contexts as well. For example, product development companies may use the method when they prepare proposals for implementation of their generic IT products for potential customers.

The IT designer is defined as the professional actor responsible for the IT design project, underscoring the importance of the element of design in such projects and, in turn, the analogy to the function of the architect in construction. The MUST method

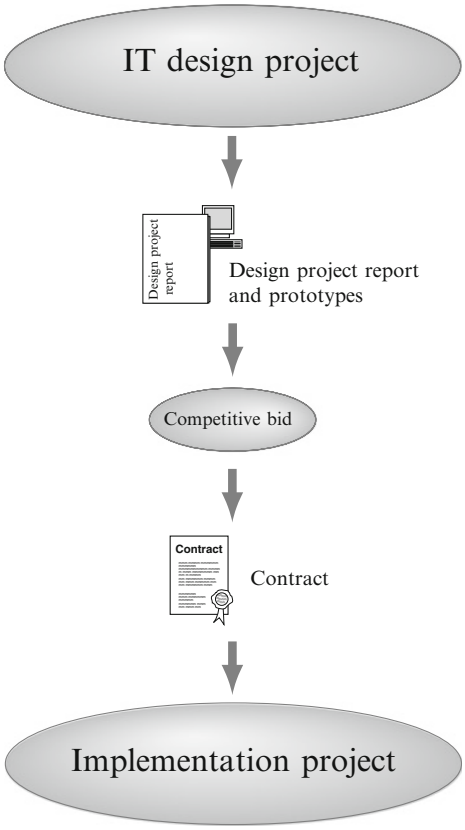

Fig. 7.2 Model of an IT project[3]

provides a perspective on IT design that takes a broad view of IT usage. This involves situating IT systems within the work organization context of which they will be part, and considering also what new qualifications will be required for users to contribute to desired changes. An important characteristic is the extensive participation of humans – that is, the management and the future users of new IT systems, along with any internal IT designers who will be participating in the implementation.

An IT design project, or design project for short, will produce a foundation for deciding how to undertake implementation projects. An IT design project is defined as a *project identifying problems, clarifying goals, and outlining solutions.* This involves:

– Analyzing the company's business and IT strategies, as well as its present goals, needs, and potentials as seen by the management and the future users of new IT systems.

[3] Bodker, Keld, Finn Kensing, and Jesper Simonsen, Participatory IT Design: Designing for Business and Workplace Realities, Figure 1.1 © 2004 Massachusetts Institute of Technology, by permission of The MIT Press.

– Designing one or more visions for overall change.
– Aligning the design visions in relation to the company's business and IT strategies.
– Setting down a strategy and plan for technical and organizational implementation, including cost estimates.
– Guaranteeing continued feedback from all relevant stakeholders.

The result of an IT design project is a report outlining one or more coherent visions for change in terms of technology, work organization, and required employee qualifications. The report may be supplemented by prototypes or (foam) mock-ups of any digital gadgets (Halse 2008). Moreover, the report includes an evaluation of the effects of implementing the visions, a cost estimate, along with a strategy and plan for implementing the visions. The report is the basis for a decision about implementation projects. This decision is typically followed by a call for tender and contract negotiations with the chosen supplier(s) that will be implementing the changes, including the IT systems.

One or several implementation projects can then be conducted in cooperation with the chosen supplier. The goal of the implementation projects is to perform the technical and organizational implementation of the chosen visions based on the design project. This will typically be done in a series of iterations, see Fig. 7.3. A vision outlines guidelines for the implementation through the documentation of goals, needs and potentials as well as through the proposed solutions. After an implementation project, or at other points in time, it is possible to evaluate achieved results in relation to the listed visions. Due to for example new solutions or problems emerging in the implementation project, parts of a

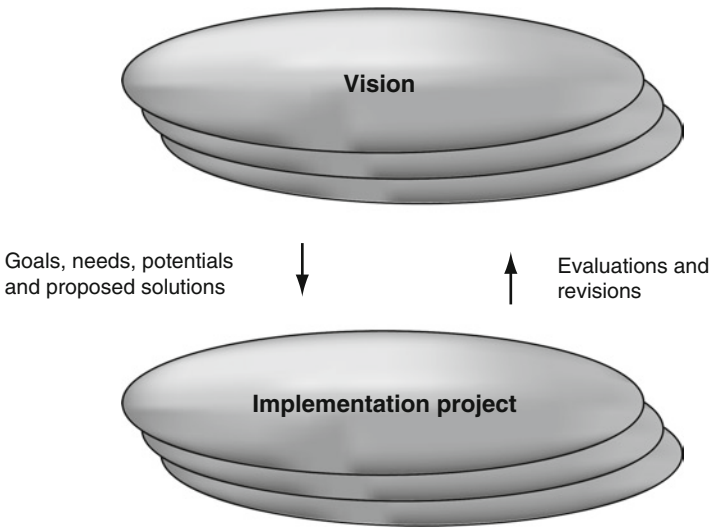

Fig. 7.3 The role of visions in a company's IT projects

vision may be revised as the result of an evaluation leading to a new foundation for succeeding implementation projects. Or an evaluation may result in a decision that a goal that was not met in the first implementation project is reformulated and set up for the next implementation project.

7.3 The Concept of 'User' in Participatory Design

A central contribution from the Scandinavian approaches to Participatory Design and Information System Development is the elaborate understanding of the user. Contrary to well known approaches like for example Contextual Design (Beyer and Holtzblatt 1998) that only talks about the 'customer', it is relevant to make several distinctions focusing not only on the relation between humans and the technology, but also to differentiate between:

- The *user*, who applies the technology for certain purposes versus the *customer* who orders and pays for it. In projects for very small companies they may be the same, but typically they are not – having sometimes very different sets of needs and goals for IT support for their work.
- *Employees* and *management*. IT design is also a political process involving conflicts and dilemmas. The resources offered by a method in terms of general principles, tools, and techniques, etc. need to reflect this.
- People having *first-, second- and third-hand knowledge* of use-processes. This entails consequences for the tools and techniques that are helpful and relevant to apply in bringing that knowledge to bear.

The premise of our participatory design method is that all types of users of a new system must be involved in different ways in the design of the relevant parts of a system.

7.3.1 User Versus Customer

Beyer and Holtzblatt (1998) present Contextual Design in which the 'customer' is defined as "anyone who uses or depends on a system." Customer is considered more inclusive than user, "which we'll use only for those who interact with the system directly" (ibid, p. 2). We may endorse the latter definition of users, while we do not talk about 'customer'. If we did, it would most likely refer to management, as customers in Beyer and Holtzblatt's vocabulary are those who pay for the system to be delivered.

When designing for workplace settings, users should explicitly be distinguished from the customer. They have different roles and competencies in organizational life in general and more specifically in IT design projects. Blurring such a distinction makes it harder for the IT designers to figure out who should be

involved in which activities, how, and for what purpose (Kensing et al. 1998b, p. 175; Bødker et al. 2004, pp. 75–79). In many projects in larger companies, users from different organizational units, or from different professional groups, will also have diverging sets of needs and requirements for IT support. In such circumstances the designer needs to be able to identify the different groups and their needs, and further to see that conflicts are dealt with and negotiated in a professional and transparent manner. Further, in larger companies it is often useful to consider also the needs of the company's external suppliers and customers, who may or may not "interact with the system directly". If they do, we would treat them as user groups. If they do not, we would treat them as important stakeholders, which again leads to other considerations about who should be involved in which activities, and for what purposes.

7.3.2 Employees Versus Management

With the distinction between employees and management the MUST method stresses the differences in terms of power and other resources available to the two groups respectively. Participatory Design (PD) have argued for user participation and suggested concrete techniques for this, as discussed by Kensing and Blomberg (1998), while IS researchers and IS methodologies traditionally have been more concerned with finding ways for IT specialists to produce basis for decisions relevant for managers.

Clement and Van den Besselaar (1993), in a review of ten PD projects, reiterate three basic requirements for participation outlined by Kensing (1983): (1) access to relevant information, (2) the possibility for taking an independent position on the problems, and (3) participation in decision making, adding two additional requirements, (4) the availability of appropriate participatory development methods and (5) room for alternative technical and/or organizational arrangements. The participation of the intended users in IT design is seen as one of the preconditions for good design. Making room for the skills, experiences, and interests of employees in system design is thought to increase the likelihood that the systems will be useful and well integrated into the work practices of the organization. Of central importance is the development of meaningful and productive relations between those charged with technology design and those who, as users, must live with its consequences. PD researchers hold that design professionals need knowledge of the actual use context, and the employees need knowledge of possible technological options. The epistemological stand of PD is that these types of knowledge are developed most effectively through active cooperation between the different user groups and designers within specific design projects.

The appraisal of which organizational members should be involved in IT design and implementation has changed over time. In the early days of PD, the central concern was to increase the participation of workers and their unions or those with little say over technological and organizational design issues affecting the workplace.

Managers rarely participated in these projects. Even today the role of management in PD projects is sometimes intentionally restricted. Some have worried that management's participation would silence the voices of employees and undermine the goal of their influence in working conditions. Bødker (1996) reports that while managers participated in some seminars and meetings during the course of an IT project, they were asked not to take part in a future workshop because their presence would make employees reluctant to express their views honestly.

Increasingly, however, people positioned throughout the organizational hierarchy (including management) and with various relations to the IT design effort are included in PD projects. Kensing et al. (1998a) report on a project in which the participation of managers, internal design professionals, and users was considered a core condition for the success of the project. Korpela et al. (1998) argue for the need to involve community members who will be served by the system under development and not solely end users. In a discussion of PD in consulting, Gärtner (1998) reports: "Customers [those funding the project] will support and pay [for the project] only if they consider risks involved to be acceptable with respect to expected outcome." In this case the involvement of the funding managers was required to secure the resources needed for the project to move forward.

7.3.3 First Versus Second and Third Hand Knowledge

First hand knowledge of a given work practice is obtained by actually performing or experiencing it. Second hand knowledge is the result of being informed of the work practice for example by interviewing the employee performing the work. Third hand knowledge is the result of for example interviewing the manager of a work group, who knows about the work from interactions with his employees. The point with this distinction is that all too often the designers, who are concerned with "getting the requirements right" or "developing a proper understanding of users' needs and concerns", rely primarily on second or third hand knowledge of the work context and daily practices that an IT system is designed to support, reorganize, substitute, or create. The reason is that it is often considered too costly or too cumbersome to become involved with the employees who actually perform the job. They may have different or even conflicting needs or interests. This is exactly why it is important to get to know what these are and why. Because needs and interests do not disappear – people will relate to new IT systems based on such needs and interests.

"Not getting the requirements right" is among the primary reasons for unsuccessful IT projects (Schmidt et al. 2001). Our position is that it is – in this light – *less* costly and cumbersome to have IT designers experience users' work practices first hand when dealt with by proper tools and techniques, and with an open attitude towards how to deal with conflicts. As described below, the principle of first hand experiences with work practices prescribe observations of users in as genuine situations as possible. This should be viewed not in contrast to, but rather as supplement to the conventional data gathering approach.

7.4 Four General Principles for Human Participation in IT Design

This section outlines four general principles for human participation in IT design that are indispensable for all participatory approaches. The principles express an overall perspective built into the MUST method and into the participatory design projects where the method is used. Applying the four principles implies a reframing of humans as compared to contemporary ISD approaches. The principles concern (1) the development of a coherent vision; (2) ensuring genuine user participation; (3) experiencing work practices first-hand; and (4) anchoring the visions with different users and stakeholders. The principles, and how they have been implemented in real-life projects, are illustrated by giving small vignettes from a number of practical cases and experiences from industry. For each principle we state the basic critical factors for achieving the principle based upon our experience.

7.4.1 The Principle of Coherent Vision for Change

A design project is carried out in a company with the aim of designing sustainable IT usage that accommodates the company's current goals and needs, enabling growth without jeopardizing its future development potential. Accordingly, IT usage should contribute to a balance of a company's resources – staff members, with their qualifications and experiences, its financial foundation and technology (including IT). For instance, the information systems should enable staff members to utilize and continue to develop their qualifications when handling their tasks. The design project should thus strive for sustainable visions, in the sense that the applications should contribute to obtaining a balance between the development, utilization, and protection of the resources of the organization.

The result of any design project is one or more coherent visions for change in the company in question and in relation to its environment. The proposed change should meet the company's revealed goals, needs, and opportunities within its business and IT strategy – which may itself need revisions as part of the project. With *coherent* we mean that the following three elements of an IT application are designed so that they each and in combination support the vision: (1) IT systems, (2) work organization, and (3) the qualifications users need to perform their job with the help of the proposed IT systems in the proposed work organization, see Fig. 7.4.

A basic, but critical, factor for this is that the customer, i.e. the person(s) commissioning the design project, accepts that these issues are dealt with in the project

Vignettte 1

> A large-scale design project was conducted in order to change the production technology at a national radio station from analogue to digital technologies (Kensing et al. 1998a). Three activities had a particularly relevance to the principle of coherent vision of change: Visiting other radio stations, developing scenarios and developing and testing prototypes.

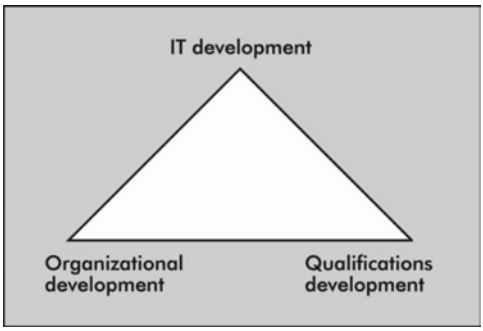

Fig. 7.4 Three elements of a coherent vision[4]

Early in the project two radio stations in Europe were visited where they used state-of-the-art digital technologies for radio production. The project group observed the radio production throughout one day and made video recordings. The video recordings documented the organization of work using the digital technologies. For example the traditional division of work between journalists and technicians was changed; journalists' selection of music was automated by using selection-programs, and parts of the program to be broadcasted during the night was prerecorded during daytime. The design visions were presented with prototypes and scenarios. The scenarios were written by the journalists who participated in the design project. In these scenarios they emphasized how management might use the new technologies to direct requests for coverage of specific events directly to the journalists and subsequently monitor the production right up to broadcasting – by having access to the material during the entire production process. Prototypes were developed with "make public" buttons illustrating how the journalists could allow management access to the material. Prototypes were distributed to all 140 employees and management at the radio station where they could interact with the proposed user interface for the envisioned systems. This way journalists and other groups could envision and to some degree experience the technology and how it might be used.

7.4.2 The Principle of Genuine User Participation

In many IT projects, the aim and focus of user participation are unclear. As a result, participation may be handled in ways that do not afford users opportunities to develop and express their needs, ideas, and visions for IT usage. If users' participation is limited to serving as informants, for instance in interviews about their work and IT needs, or to taking part in systems testing, or if participation is limited to middle

[4] Bodker, Keld, Finn Kensing, and Jesper Simonsen, Participatory IT Design: Designing for Business and Workplace Realities, Figure 2.1 © 2004 Massachusetts Institute of Technology, by permission of The MIT Press.

managers or executives, who may excel at representing the company's overarching goals for the project there is little chance that the resulting systems cover the users' actual needs. Such projects lacks the insight into day-to-day routines, including what factors may be complicating the work or what useful alternatives could be.

The principle of genuine user participation calls for the active participation of user representatives influencing the process of design as well as the visions it results in. There are two rationales for this, a pragmatic and a political. The pragmatic arguments rest on the need for mutual learning between users and IT designers: IT designers need knowledge about the work environment that is the domain of the design project, and users need knowledge about technological options. That end is most effectively attained by organizing activities that enable the two groups to learn from each other. Users can contribute innovative and constructive suggestions for change when they have the right conditions for doing so. The political arguments revolve around the users' right to influence their own working conditions, which are often significantly affected by IT projects. Modern companies rely on professionals, i.e. highly skilled, autonomous, and knowledgeable employees. Consequently, it is a core HR strategy to ensure staff members' influence on their own working conditions and working environment in order to keep (and attract) competent and ambitious staff members.

One should also note that in most real-life projects the user community is so large and diverse that the IT designer must rely on *representatives* of users to be involved in a design project. Thus, the selection of representatives becomes a crucial element in the design project, where compromises have to be made. Often there is a wish to include experienced persons capable of thinking out-of-the-box who are also well respected by their colleagues. However, in the daily business these people are busy and in short demand.

Genuine user participation increases the potential of visions produced by a design project to reflect the users' true situation and needs. And it further increases the potentials of the systems to be used according to their intentions. A basic critical factor is that sufficient time and resources are set aside for this (Clements and Van den Besselaar 1993).

Vignette 2

A design project was conducted by a large IT vendor to improve clinicians' overview of patients at intensive care units. A series of workshops were conducted where nurses and physicians from three intensive care units participated in designing the system. During these workshops, the physicians repeatedly stressed the importance of having graphical representations of results from the ongoing samples and measurements (blood pressure, oxygen levels, temperature, pulse rate, etc.). Much effort was directed to the design of different types of graphs, their colors, forms, configurability, etc. These representations constituted a major resource of developing the system. When the system was implemented and evaluated it was observed that the graphical representations were almost never used. It turned out, that the practical measurements and on-going monitoring of the patients' vital parameters was conducted by the nurses – and they did not use or appreciate graphical representations. Instead they measured and recorded the actual measurements and used the actual figures on the forms and charts to maintain the status of their overview of the patients. The physicians' statements represented second-hand knowledge of the work processes

conducted by nurses. Due to their different ranks in the hierarchy, the nurses did not object to the design requirements voiced by the physicians. From this project the vendor learned that they had to claim an influence on who the customer put at the disposal for the design workshops and to insist on having users with first-hand knowledge of the work processes in question present – without they are compromised by second-hand or third-hand users with higher rank or management-like status.

7.4.3 The Principle of First Hand Experience with Work Practices

There are three different ways, basically, of obtaining new insight into subject matters relevant to a design project. You can read up on the subject, you can ask knowledgeable people to tell you about it and, finally, you can put yourself in a situation where you experience the subject first hand. The first two ways are most commonly used in ISD methods in general, and they reflect second-hand or third-hand knowledge. This implies descriptions of work in terms of processes and procedures reflecting an ideal work flow. This is different from how the work is actually carried out as pointed out by Suchman: Users act in a situation and do not follow plans and procedures in any narrow sense (Suchman 1983, 1987). Thus, there is a need to also include first hand experiences. A major tenet of this principle is that work is a socially organized activity where the actual behavior differs from how it is described, prescribed, or envisioned.

Ethnographically inspired observations are the primary means to realize the principle. In a design context the aim of ethnography is to develop a thorough understanding of work practices as a basis for the design of IT systems (Simonsen and Kensing 1997, 1998; Simonsen 2009). Using ethnography in design has been acknowledged especially within the fields of Participatory Design, e.g. (Greenbaum and Kyng 1991; Schuler and Namioka 1993; Bødker et al. 2004) and Computer Supported Cooperative Work (CSCW), e.g. (Sommerville et al. 1992; Hughes et al. 1993; Luff et al. 2000).

The principle of first hand experience implies that studies of work practices must include observations, possibly supported by video analyses. The work practices include existing work practices (users performing their regular business prior to system implementation) or new envisioned work practices (users trying out prototypes in situations as realistic as possible, or visits to other companies where similar technologies are in use). An obvious critical factor for this principle is to get access to the relevant work practices. Often this includes negotiations with management and/or the work groups to establish the conditions. In some design projects however, there is no current work practice, i.e. in projects dealing with radical new products or in new domains. In such projects we may have to “create the work practice” by simulating work and IT tools by prototyping and enacting scenarios.

Vignette 3

A design project included the design of IT support for the production manager and for the editors from the Danish Film Institute (Simonsen and Kensing 1997). From the beginning,

it was voiced that "everybody should be able to see all information in the system." After we had observed the editors for some time, they became confidential with us and suddenly – at a follow up interview – one of them entrusted in us that there was a (legitimate but manifest) conflict between the production manager and the editors who all had a budget for their area: complete openness of all information in the system would favor the production manager and weaken the editor's influence in the organization. For example, financial support of productions considered by the editors should be strictly confidential. None of the editors' personal calculations (about which productions they were considering to fund and with how much) should be public unless made so by the editor managing the production. If this part of the system was open to all, the editors simply would not use it for this complex task. We had to carefully contemplate how to bring up this issue without taking part in the conflict. We decided to present two alternative design proposals: One implicitly in favor of the production manager and one explicitly supporting the editors, who had confided in us. At a steering committee meeting the proposal supporting the editors was chosen – though not without controversies.

7.4.4 The Principle of Anchoring Visions

The anchoring principle means ensuring that stakeholders understand and support the design project's goals, visions, and plans. Involving and informing relevant stakeholders is a key issue in ISD, especially with regard to top management and the staff of involved sections. Top management involvement and the development of strong relationships with top management continue to be reported as the uttermost important challenge within IS, see for example (Xia and Lee 2004) and (Schmidt et al. 2001).

In a design project, the project group develops various representations of the existing situation and visions of desired changes. Such representations are perceived and interpreted individually and differently by the people to whom they are presented. An important means of anchoring is to communicate representations that provide the most coherent image, as interpreted by the project group, to other relevant stakeholders that are not directly involved in the project. The principle of anchoring visions focuses on three stakeholder groups: (1) (top) management, who has the power to decide whether or not the proposed visions will be implemented, (2) employees and other interested parties, who will either use the IT systems in question or be effected by them, and (3) internal and external groups that at a later stage become involved in the technical and organizational implementation of the proposed visions.

Anchoring visions encompasses informing about and promoting understanding and backing for the relevance of the design project and its goals and visions. This includes inviting stakeholders to discuss, review, challenge, and reformulate the project groups' arguments for how a specific IT-based proposal solves an experienced problem or supports an important business need. The principle prescribes that stakeholders must be informed and involved in various ways to be able to evaluate the consequences of the proposed changes, as seen from each of their perspectives. This needs to occur in time for the project group to incorporate their reactions into the final design proposals.

To achieve the envisioned changes, continuity is a critical factor. This is why it is important to anchor the visions with the staff of involved departments and sections as well as with (top-) management. This is also why it is important to have central actors from the design project take part in the implementation and/or have central persons from the implementation team involved in the anchoring activities of the design project.

Vignette 4

A large international vendor of an enterprise system conducted a design project with a potential customer in order to clarify if the customer was in a situation where he could benefit from implementing (major parts of) the system (Simonsen 2007). Thirteen selected employees representing different areas of the customer's organization and business processes were interviewed. The vendor's IT designers were very experienced within the business domain of the customer. Based on the interviews they were able to develop a convincing generalization of the situation, which identified and characterized relevant problem domains. Their image of the situation was tied together by a string of assumptions and hypotheses generalizing the information gathered from the interviews. The IT designers presented the results of their interviews at a full day workshop for the customer's top-management including the CEO, CFO, and CIO. They systematically went through each argument chain relating identified problems or needs with proposed IT solutions specifying each problem or need, identifying its causes, the (undesirable) consequences it led to, and the ideas for its solution. The workshop acquainted the management group with the IT designer's analysis and diagnosis of the current state of affairs and involved them in a structured discussion of each line of argument related to an identified problem or need, and invited them to challenge central suppositions, assumptions, and hypotheses related to the causal relation between problems and solutions. The workshop disclosed the IT designer's experience and knowledge within the business domain as well as the customer's specific context and situation. In this way, it provided the customer management confidence in the vendor's competence and in the relevance of the proposed IT solutions, and hereby it anchored the visions.

7.5 Comparison with Other Methods

The issues and concerns dealt with by the MUST method are also addressed by other contemporary ISD approaches. This section outlines significant similarities and differences with regard to agile methods, RUP, Contextual Design, BPR, and Lean.

Contemporary approaches have abandoned the classic waterfall model with predefined phases. Rapid Application Development, or "agile" methods like Extreme Programming, focuses on fast deliveries of potentially operative systems and incremental development, relying on project models with strong iterative elements controlled along the time dimension by time boxes. These approaches differ in various ways, but they share a strong focus on programming and implementation aspects. A basic assumption for a project following these approaches is that a decision to build a system of a particular kind has already been made. In contrast to this, we specifically propose an upfront design project to establish the foundation for such a decision. Also, these methods are not intended for larger projects involving multiple systems, some of which are customized systems integrated within an existing system

portfolio. Methods like modern object-oriented software engineering methods such as Rational Unified Process (RUP), focus on building systems from scratch. RUP, in turn, does incorporate early design activities – in the inception and elaboration phases. These activities are integrated into a software engineering method with a strong focus on modeling, specifications, and implementation, striving for the classic virtues of robustness and maintainability.

Contextual Design (CD), as formulated by Beyer and Holtzblatt (1998) and Holtzblatt et al. (2004), has a scope similar to MUST referred to as front-end design, requirement engineering, or systems analysis. However, CD does not distinguish between the users – those that will interact directly with the systems – and the customers – those that order and pay. In addition, the method does not suggest ways of handling potential conflicting interests. An IT design project may involve politics and we must be explicit about the different roles and competencies in organizational life in general and in IT design projects in particular. While a CD process aims at specifications meant for developers or coders, including detailed object oriented models of the system functionality and structure, the MUST method involves a separate design project where such specifications are deferred until a decision has been made on what to build or buy. In this way, MUST is inscribed in an overall project model where it is assumed that not all IT systems are built from scratch and where the implementation of customized systems will most likely be outsourced. The rationale of CD seems to be that the same group of people proceeds all the way to implementation, in which case this type of detailed description is valuable. But detailed technical descriptions are superfluous for those systems that the company in question decides to buy as standard systems, those that are outsourced for a vendor to deliver, and those that are decided not to be pursued any further.

Business Process Reengineering (BPR), in its original form as proposed by Hammer and Champy (1993), has the same scope as the MUST method. Both address the early analysis and design activities in an IT design project as well as project management. Both aim at formulating one or more visions for the future use of IT, while the technical and organizational implementation is considered outside the scope of these methods. BPR and MUST consider the relations between a design project and an organization's business and IT strategies, which are either neglected or considered outside the scope of many current methods – with potentially damaging results. While radical change, including downsizing, is a major part of the rationale of BPR, it does not deal with ethical or practical issues in relation to users. MUST states explicitly that if management aims at job cuts or other drastic changes, this should be announced up front. If users know and accept these objectives, we still recommend a participatory approach. Instead BPR suggests an expert strategy, neglecting the knowledge, experience, and interests of users, thereby risking that the visions developed do not meet real needs. BPR orients its deliverables primarily toward management, offering no help in understanding, developing, or presenting relations between IT and users' work practices. The content and the form of the reports and prototypes resulting from a MUST process are meant for management to prioritize further directions for the subsequent implementation activities. They also allow users to understand the consequences – as to their work practices – of the proposed coherent visions for change.

Lean represents an approach partly associated with BRP. Lean is a management philosophy originally developed at the Toyota Corporation and Lean is sometimes referred to as the "Toyota Way". The method's application area is thus by origin manufacture, but according to Womack and Jones (1996) the method is generally applicable to organizational innovation and change processes. As the name implies, the idea is to make processes 'lean' by removing or reducing all activities that are not producing value for the customer. Even though the objective of design is to create something new, MUST incorporates 'lean-thinking' by establishing the objective of the design project in relation to the context of the company as well as other ongoing projects early in the project. MUST does not stipulate how and where to lean the processes; instead the aim of the general principles for example on how to involve the human actors, the methodological guidelines and the techniques and tools is to create sustainable solutions.

7.6 Implications

This concluding section discusses implications and potential further directions when participatory design is taken out of the "research lab" and is applied in real life settings. First some of the lessons learned from assisting IT practitioners integrating PD into their work practice, and from teaching PD to students, are reflected upon. Then we conclude by briefly outlining some directions from our current research related to the MUST method, where it is applied in new contexts.

The MUST method, of which the four general principles and some of its main ideas have been described in this chapter, has been developed as part of a research program organized around 14 projects in Denmark and the US. Over more than 10 years we have cooperated with private and public organizations in the development of the method. Further, as the method was developed in an explorative and incremental way, our undergraduate and graduate students tried out various elements of the method in their course work and master thesis projects – some of which were also carried out in cooperation with external partners. Thus, dissemination activities were conducted hand in hand with the explorative and incremental development of the method.

7.6.1 Bringing PD to IT Practitioners

Integrating new methods in established work practices is difficult and therefore the introduction of new methods often fails (Bansler and Bødker 1993). However, it is indeed possible for IT practitioners to change their work practice and start using the MUST method (Kensing 1999, 2003), (Bødker et al. 2002, 2004). A short introduction (1 or 2 days) may work as a kick-off workshop for starting using the method – but this has to be supplemented. An approach to method dissemination must be based on two basic premises:

1. Introduction of a new method should be coupled with a joint appreciation of actual challenges in real design projects.
2. Traditional teaching cannot stand alone in method dissemination.

These premises have emerged from numerous projects in collaboration with IT practitioners. A successful dissemination process should comprise a combination of lectures, reflections on current and emerging practices, apprenticeship relations, and supervision of technical skills as well as personal competences. The central point is to get beyond a mode of detached reflection in the interaction between the IT practitioners and the person responsible for the dissemination endeavor (in our case, us as researchers).

Practitioners who are simply given a general presentation of a new technique are left on their own when trying to integrate the technique into their work practices. And a disseminator who is simply told about events and changes in a recent project is left with the question about what really happened.

So, to get beyond this problem (of second- and third-hand knowledge), the disseminator must get involved in the work of the IT practitioners through observations or ultimately through working together on a project. This makes it possible for the disseminator to relate to problems in the practitioners' current practices when presenting a new technique or proposing changes in their design practice.

7.6.2 Bringing PD to Students

The Danish version (Bødker et al. 2000, 2008) and English version (Bødker et al. 2004) of the MUST book has been used as the primary textbook in introductory design courses for graduate students in IS. The general format of these courses is designed based on the premise that students need to practice and develop these skills in order to learn to master the elements of participatory design. Half of the course is traditional lectures by the professor, whereas the other half is devoted to a project assignment. Here students work in groups of two to four persons on a project where they have to solve a real world IT design task. The students are asked to engage with a small company, a public institution, a non-profit organization, or a department or section within a larger corporation. Below, two lessons are presented: the first related to the structure of the course, the second related to what is learned by the project work.

Students enter the course with prerequisites in programming and requirements modeling from SE courses. In an introductory course, students need some kind of 'structure' to guide their learning process from reading about the method to start practicing a situation-specific combination of the resources provided, as depicted in Fig. 7.1. The guiding structure that has proven most efficient is the organization of a design project into separate activities.

The first part of the lectures is organized as a step by step walkthrough of a design project. The lectures highlight which techniques were chosen in order to follow each of the four principles and to help meet the requirements set for the results of

each of the phases. This structure is also recommended to guide the students' first PD-project. However, when supervising subsequent PD-projects, students should be advised to also include their appreciation of the design situation at hand to inform the ways in which they combine the resources provided by the method. Even though the method includes guidelines for how and when for example to reduce a phase into an activity in the preceding phase, it takes more experience to master these types of decisions.

7.6.3 Ongoing Research

Research in relation to PD and the MUST method is still ongoing. These years we investigate, elaborate, and expand our approach within two directions: (1) How to manage participatory approaches applied throughout the design and organizational implementation of especially large-scale systems, referred to as a 'sustained' participatory design (Simonsen and Hertzum 2008, 2011), and (2) supporting communication and collaboration across organizational boundaries and between organizational members and individuals (patients), (www.cith.dk). We conduct experiments in an explorative and experimental way working with (close to) real life projects, and we expect this to lead to modifications and clarifications of various elements of the method and an evaluation of how it fares in the new contexts.

References

Ballejos, L. C. & Montagna, J.M. (2008) Method for stakeholder identification in interorganisational environments. *Requirements engineering* 13, 281–297.

Bansler, J and Bødker, K. (1993) A Reappraisal of Structured Analysis: Design in an Organizational Context. *ACM Transactions on Information Systems,* 11(2), 165–193.

Beyer, H. & Holtzblatt, K. (1998) *Contextual Design. Defining Customer-Centered Systems*. Morgan Kaufmann Publishers, Inc, San Francisco, CA.

Bødker, S. (1996) Creating Conditions for Participation: Conflicts and Resources in in Systems Development. *Human-Coomputer Interaction,* 11(3), 215–236.

Bødker, K., Kensing, F. & Simonsen, J. (2000) *Professionel IT-forundersøgelse – grundlaget for bæredygtige IT-anvendelser*. Samfundslitteratur

Bødker, K., Kensing, F. & Simonsen, J. (2002) Changing Work Practices in Design. In *Social Thinking – Software Practice* (Eds, Dittrich, Y., Floyd, C. & Klischewski, R.) MIT Press, Cambridge, MA, pp 267–285.

Bødker, K., Kensing, F. & Simonsen, J. (2004) *Participatory IT Design. Designing for Business and Workplace Realities*. MIT press, Cambridge, MA.

Bødker, K., Kensing, F. & Simonsen, J. (2008) *Professionel IT-forundersøgelse – grundlag for brugerdrevet innovation (2 udg.)*. Samfundslitteratur

Clement, A. & Besselaar, P.V.d. (1993) A Retrospective Look at PD Projects. *Communications of the ACM*, 36, 29–37.

Coughlan, J. & Macredie, R.D. (2002) Effective Communication in Requirements Elicitation: A Comparison of Methodologies. *Requirements Engineering* 7, 47–60.

DeMichelis, G., Dubois, E., Jarke, M., Matthes, F., Mylopoulos, J., Schmidt, J. W., Woo, C. & Yu, E. (1998) A three-faceted view of information systems, *Communications of the ACM*, 41(12), 64–70.

Gärtner, J. (1998) Participatory Design in Consulting. *Computer Supported Cooperative Work – A Journal of Collaborative Computing*, 7(3–4), 273–289. Kluwer Academic Publishers, Dordrecht, The Netherlands.

Greenbaum, J. & Kyng, M. (Eds.) (1991) *Design at Work: Cooperative Design of Computer Systems* Lawrence Erlbaum Associates, Chichester, UK.

Halse, J. (2008) *Design Anthropology: Borderland Experiments with Participation, Performance and Situated Intervention*. Ph.d. dissertation. IT University of Copenhagen, Copenhagen, 2008.

Hammer, M. & Champy, J. (1993) *Reengineering the Corporation. A Manifesto for Business Revolution*. HaperBusiness/HaperCollins, New York.

Holtzblatt, K., BurnsWendell, J. & Wood, S. (2004) *Rapid Contextual Design: A How-to Guide to Key Techniques for User-Centered Design*. Morgan Kaufmann Publishers Inc, San Francisco, CA.

Hughes, J.A., Randall, D. & Shapiro, D. (1993) From Ethnographic Record to System Design: Some Experiences From the Field. *Computer Supported Cooperative Work (CSCW): An International Journal*, 1, 123–141.

Iivari, J., Isomäki, H., and Pekkola, S., (2009). The User – the Great Unknown of Systems Development: reasons, forms, challenges and intellectual contributions of user involvement, *ISJ special issue introduction*, to be published.

Kensing, F. (1983). The Trade Unions Influence on Technological Change. In *Systems Design For, With and By the Users*. U. Briefs et al. (eds.), North Holland Publishing Company, New York.

Kensing, F. and J. Blomberg (1998) PD meets CSCW – Issues and Concerns. *Computer Supported Cooperative Work*, 7, 167–185.

Kensing, F. (1999). Method Design and Dissemination. In J. Pries-Heje et al. (eds.): Proceedings of The Seventh European Conference on Information Systems, Copenhagen, Denmark.

Kensing, F., Simonsen, J. & Bødker, K. (1998a) Participatory Design at a Radio Station. *Computer Supported Cooperative Work*, 7, 243–271.

Kensing, F., Simonsen, J. & Bødker, K. (1998b) MUST – a Method for Participatory Design. *Human-Computer Interaction,* 13(2), 167–198.

Kensing, F. (2003). Methods and Practices in Participatory Design. ITU Press.

Korpela, M., H.A. Soriyan, K.C. Olufokunbi, A.A. Onayade, A. Davies-Adetugbo & D. Adesanmi (1998): Community Participation in Health Informatics in Africa: An Experiment in Tripartite Partnership in Ile-Ife, Nigeria. *Computer Supported Cooperative Work – A Journal of Collaborative Computing*, 7(3–4), 339–358. Kluwer, Norwell, MA.

Luff, P., Hindmarsh, J. & Heath, C.C. (2000) *Workplace studies: recovering work practice and informing system design*. Cambridge University Press, Cambridge.

Schmidt, R., Lyytinen, K., Keil, M. & Cule, P. (2001) Identifying software project risks: An international Delphi study. *Journal of Management Information Systems*, 17, 5–36.

Schuler, D. & Namioka, A. (Eds.) (1993) *Participatory Design: Principles and Practices* Lawrence Erlbaum Associates, London, UK.

Simonsen, J. (2007) Involving Top Management in IT Projects: Aligning Business Needs and IT Solutions with the Problem Mapping Technique. *Communications of the ACM*, 50, 53–58.

Simonsen, J. (2009) The Role of Ethnography in the Design and Implementation of IT Systems. *Design Principles and Practices, an International Journal*, 3(3), 251–264.

Simonsen, J. & Hertzum, M. (2008) Participatory Design and the Challenges of Large-Scale Systems: Extending the Iterative PD Approach. *Proceedings of the 10th anniversary conference on Participatory Design: Experiences and Challenges, September 30–October 4, 2008, Bloomington, IN, USA*, pp 1–10.

Simonsen, J. and M. Hertzum (2011): "Iterative Particpatory Design", in Simonsen, J., J.O Bærenholdt, M. Büscher, and J.D. Scheuer (Eds.) *Design Research: Synergies from Interdisciplinary Perspectives,* Routledge, London.

Simonsen, J. & Kensing, F. (1997) Using Ethnography in Contextual Design. *Communications of the ACM*, 40, 82–88.

Simonsen, J. & Kensing, F. (1998) Make Room for Ethnography in Design! *The Journal of Computer Documentation*, 22, 20–30.

Sommerville, I., Rodden, T., Bentley, R. & Sawyer, P. (1992) Sociologists can be surprisingly useful in interactive systems design. *Proceedings of HCI'92, York University, September 1992, People and Computers*, VII, 341–353.

Suchman, L.A. (1983) Office Procedure as Practical Action: Models of Work and System Design. *ACM Transactions on Office Information Systems*, 1, 320–328.

Suchman, L.A. (1987) *Plans and Situated Actions: The Problem of Human-Machine Communication*. Cambridge University Press, Cambridge, NY.

Womack, J.P. & Jones, D.T. (1996) *Lean Thinking: Banish Waste and Create Wealth in Your Corporation*. Simon and Schuster, London.

Xia, W. & Lee, G. (2004) Grasping the Complexity of IS Development Projects. *Communication of the ACM*, 47, 69–74.